Skip BCH4 User Manual

General Description and Name

This scheme Implements calculate ECC with BCH4. But there is a little different that output has 4 bits shift left than standard BCH4.
During programming handle the same as 'Skip bad block' (bad block will be skipped within partition)

Relevant User Options

The following special features on the special features tab apply to this scheme. The default values might work in some cases but please make sure to set the right value according to your system.
Please note only the below special feature items are related to this scheme and ignore any others. If any of below items doesn't exist, please check whether the right version has been installed or contact Data I/O for support by submitting Device Support Request through this address:
http://www.dataio.com/support/dsr.asp

Bad Block Handling Type = "Skip BCH4"

Spare area = Normally "ECC"
Or depend on your data file:

Enabled ➔ Data file contain spare.
Disabled➔ Data file does not contain spare, Task Link will fill with blank status to spare area.
ECC ➔ Data file does not contain spare, Task Link will fill spare with blank status then overwrite correct ECC into spare.

ECC Extended Type = same as above 'Spare area'

PartitionTable File : Point to a .mbn file which describes the partition information.

Special Notes

Format of PartitionTable.mbn:

a. Binary file fixed length 256 bytes.
b. Organization:16 rows x 4 columns. Each table item is 32-bits, little endian byte ordering.
c. Each row of the table describes configuration for one partition. Up to 16 partitions can be used.
d. Partition configuration:
    i. **Start Adr**: address of start of partition in flash blocks. The programmer will set the file read pointer and the programmer write pointer to Start Adr. If Start Adr=0xFFFFFFFF, skip to the next partition.

ii. **End Adr**: last valid block in the current partition. The last data block programmed must be equal to or less than End Adr, otherwise the programmer will reject the flash device.

iii. **Actual Data Length**: number of blocks of data to read from the input file and write to the flash in the current partition

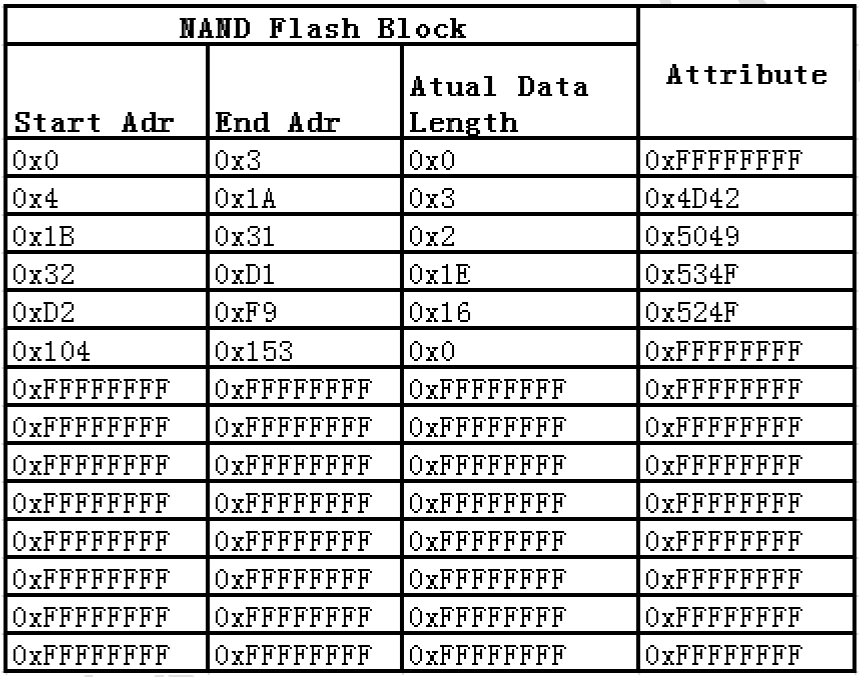
1. For partition SBT, actual data length is normally set to ZERO.

iv. **Attribute**: don’t care for this part. Normally leave as blank 0xFFFFFFFF

Please note to keep: **Actual Data Length + max bad blocks allowed <= End Adr - Start Adr + 1**

v. Example PartitionTable.mbn file:

41475.mbn



NAND Flash Block			Attribute
Start Adr	End Adr	Atual Data Length	
0x0	0x3	0x0	0xFFFFFFFF
0x4	0x1A	0x3	0x4D42
0x1B	0x31	0x2	0x5049
0x32	0xD1	0x1E	0x534F
0xD2	0xF9	0x16	0x524F
0x104	0x153	0x0	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF

Revision History

V1.0 Date: 2015-03-27
Create this spec.

V1.1 Date: 2015-04-22
Added some special feature and describe for MBN table.

V1.1 Date: 2015-05-13
Update description.

Appendix

You can get the file “Description of common NAND special features.pdf” from http://ftp.dataio.com/FCNotes/BBM/